UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,806	12/26/2001	Thomas J. Bonola	1662-49800 JMH (P98-2417)	6235

			EXAMINER	
23505	7590	01/29/2004	INOA, MIDYS	

CONLEY ROSE, P.C.
P. O. BOX 3267
HOUSTON, TX 77253-3267

ART UNIT	PAPER NUMBER
2188	

DATE MAILED: 01/29/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/037,806	BONOLA, THOMAS J.
	Examiner	Art Unit	
	Midys Inoa	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *26 August 2002*.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-38* is/are pending in the application.

   4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-38* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on *26 December 2001* is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a) ☐ All b) ☐ Some * c) ☐ None of:

   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____.
   3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

   a) ☐ The translation of the foreign language provisional application has been received.

14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *4* .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

Information Disclosure Statement

1.   The information disclosure statement (IDS) submitted on August 26th, 2002 has been

considered by the examiner.

Drawings

2.   The drawings filed on December 26th, 2001 have been accepted by the examiner.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed
> in the United States before the invention by the applicant for patent or (2) a patent granted on an application for
> patent by another filed in the United States before the invention by the applicant for patent, except that an
> international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this
> subsection of an application filed in the United States only if the international application designated the United
> States and was published under Article 21(2) of such treaty in the English language.

4.   Claims 1-38 are rejected under 35 U.S.C. 102(e) as being anticipated by Traiman et al.

(Us 2002/0144073 A1).

Regarding Claims 1-2, 7, Traiman et al. teaches a memory heap in which a memory block

from the heap is allocated for use and removed from the linked list of free blocks ("heap pile" or

"linked list heap memory management system", Page 3, paragraph 36). Once the allocated block

is no longer needed, it is freed and returned to the linked list, thus becoming the first free block

available for use (Page 3, paragraph 38). Since the newly freed block is being placed as the first

block available for use, it is understood that its placement is at the beginning of the list, which is

also one of the ends of the linked list.

Regarding Claim 3, Traiman et al. discloses a method for placing a freed block of heap memory at the beginning of a linked list. In addition, Traiman discloses that the same method, only modified can be used to place the block of heap memory at the end of the linked list. Therefore, it is understood that in order to perform this task, the new last block must have a "null" pointing to the "next free block address" field, the former last block in the linked list must now point to the new last block as its "next free block address", and the new last block in the linked list must point to the former last block in the linked list as its "previous free block address" (Page 3, paragraph 38).

Regarding Claim 4, Traiman et al. discloses freeing the allocated block when its no longer needed and returning it to the linked list, thus becoming the first free block available for use (Page 3, paragraph 38). Since the newly freed block is being placed as the first block available for use, it is understood that its placement is at the beginning of the list.

Regarding Claim 5, Traiman et al. discloses the method for placing a freed block of heap memory at the beginning of a linked list as being setting the "previous free block address" of the former first block in the list to point to the new block, setting the "next free block address" of the new block to point to the former first free block, and setting the "previous free block address" of the new block to "null" (Page 3, paragraph 38)

Regarding Claim 6, it is understood that prior to setting the new block's "next free block address" to that of the former first block, one must know what that address is, and reading it is the only way to acquire it (Page 3, paragraph 38).

Regarding Claim 8, since the first free block available for use in the linked list is placed at the beginning of the list, it is understood that when a memory block is allocated and removed

from the linked list, this block of heap memory must come from the beginning of the linked list (Page 3, paragraphs 38-39).

Regarding Claim 9, Traiman et al. discloses checking to see if the next block field is a "null address" prior to removing a block of heap memory from the linked list. Since null address means that no other block of the specified size exists within the heap, another search is started for a bigger block of heap memory. If the next block field is not a "null address" then, the block of heap memory is allocated and removed from the linked list (Page 3, paragraph 37).

Regarding Claims 10-11, since the first free block available for use in the linked list is placed at the beginning of the list, it is understood that when a memory block is allocated and removed from the linked list, this block of heap memory must come from the beginning of the linked list and therefore, when searching for a block of heap memory to allocated, it is the first register at the beginning of the linked list that is first read (Page 3, Paragraph 38-39). In addition, when the top block of the heap memory is removed, the new top block must be adjusted to point to the top of the heap.

Regarding Claims 12-13, Traiman et al. teaches a memory heap in which a memory block from the heap is allocated for use and removed from a linked list of free blocks ("heap pile" or "linked list heap memory management system", Page 3, paragraph 36). Once the allocated block is no longer needed, it is freed and returned to the linked list, thus becoming the first free block available for use (Page 3, paragraph 38). Since the newly freed block is being placed as the first block available for use, it is understood that its placement is at the beginning of the list, which is also one of the ends of the linked list. Traiman et al. discloses a method for placing a freed block of heap memory at the beginning of a linked list. In addition, Traiman discloses that the same

method, only modified can be used to place the block of heap memory at the end of the linked list. Therefore, it is understood that in order to perform this task, the new last block must have a "null" pointing to the "next free block address" field, the former last block in the linked list must now point to the new last block as its "next free block address", and the new last block in the linked list must point to the former last block in the linked list as its "previous free block address" (Page 3, paragraph 38).

Regarding Claims 14-15, Traiman et al. discloses the method of removing a block from the linked list when it has been allocated for use. It is understood, that once the top block is allocated, the next block becomes the new top block. Therefore, when the top block of the heap memory is removed, the new top block must be adjusted to point to the top of the heap (Page 3, paragraph 37).

Regarding Claim 16-18, Traiman et al. discloses the method for placing a freed block of heap memory at the beginning of a linked list as being setting the "previous free block address" of the former first block in the list to point to the new block, setting the "next free block address" of the new block to point to the former first free block, and setting the "previous free block address" of the new block to "null" (Page 3, paragraph 38)

Regarding Claims 19-21, Traiman et al. teaches a memory heap in which a memory block from the heap is allocated for use and removed from the linked list of free blocks ("heap pile" or "linked list heap memory management system", Page 3, paragraph 36). Once the allocated block is no longer needed, it is freed and returned to the linked list. Items placed at the beginning of the linked list are the first free blocks available for use (Page 3, paragraph 38). Traiman discloses the ability of placing the returned heap memory blocks at the end of the linked list as

well as at the beginning. In placing them at the end, the newly released items would be the last

free blocks available for use. In this manner, the system would be operating in a LIFO fashion

(Page 3, Paragraph 36-38).

Regarding Claims 22-25, Traiman et al. discloses the method of removing a block from

the linked list when it has been allocated for use. It is understood, that once the top block is

allocated, the next block becomes the new top block. Therefore, when the top block of the heap

memory is removed, the new top block must be adjusted to point to the top of the heap (Page 3,

paragraph 37).

Regarding Claims 26-28, Traiman et al. discloses a method for placing a freed block of

heap memory at the beginning of a linked list. In addition, Traiman discloses that the same

method, only modified can be used to place the block of heap memory at the end of the linked

list. Therefore, it is understood that in order to perform this task, the new last block must have a

"null" pointing to the "next free block address" field, the former last block in the linked list must

now point to the new last block as its "next free block address", and the new last block in the

linked list must point to the former last block in the linked list as its "previous free block

address" (Page 3, paragraph 38).

Regarding Claim 29, Traiman et al. teaches a memory heap within a main memory which

is part of a computer system where the computer system is known to include a microprocessor, a

bridge to couple the microprocessor and main memory, hardware devices, and a graphics card,

amongst other components. In the heap memory, memory blocks from the heap are allocated for

use and removed from a linked list of free blocks ("heap pile" or "linked list heap memory

management system", Page 3, paragraph 36). Once the allocated block is no longer needed, it is freed and returned to the linked list (Page 3, paragraph 38).

Regarding Claim 30, Traiman et al. teaches heap memory blocks that could be of the same or could be combined to form larger blocks of heap memory (Page 3, paragraph 39). In addition, the blocks of heap memory contain headers each having a next block field, and a previous block field (Page 3, paragraph 34). The free blocks of heap memory are managed in a linked list and the blocks are linked by the pointers in the headers (Page 3, paragraph 35).

Regarding Claims 31-32, the pointers in the headers of the heap memory blocks form a linked list of free blocks, which has a beginning and an end. Those block placed that the top of the linked list are the next available for use ("removes blocks from the top of the linked list", Page 3, paragraph 34-38). In addition, Traiman et al. discloses the ability of placing newly freed blocks at the top of the linked list or at the bottom (Page 3, paragraph 38).

Regarding Claims 33-38, Traiman et al. discloses allocating memory blocks in the heap for use in any task involving any of many hardware devices within a computer system. Therefore, the hardware device in question could be a graphics card, a network card, an audio card, a hard drive, or any other storage device or computer component (Page 3, paragraph 33).

Conclusion

5.   The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- Bonola (US 6,412,053), System Method and Apparatus for Providing Linearly Scalable Dynamic Memory Management in a Multiprocessing System"

- Meier (6,076,151), Dynamic Memory Allocation Suitable for Stride-Based Prefetching

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Midys Inoa whose telephone number is (703) 305-7850. The

examiner can normally be reached on M-F 7:00am - 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Mano Padmanabhan can be reached on (703) 306-2903. The fax phone number for

the organization where this application or proceeding is assigned is (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding

should be directed to the receptionist whose telephone number is (703) 305-3900.

Midys Inoa
Examiner
Art Unit 2188

MI

Mano Padmanabhan
1/26/04

MANO PADMANABHAN
SUPERVISORY PATENT EXAMINER
TC2100